

Sprint 4 Plan - EasyTrage, TeamTrage

Sprint completion date 3/18/18

Revision 2, Revision Date 3/13/18

Red = unmet goal, blue = revision

Goals: Implement user account settings controls, enable automated trading for each coin and implement functionality to view past trades

Tasks:

1. (8 story points) User story 0: As a user, I want to be able to edit my account settings and input my API keys of my exchange accounts
 - a. Make 'Settings' component with a form linked from the Dashboard where users can edit fields pre-populated with their current settings data (2-4 hours)
 - b. Make settings route on /api/auth route on the backend to accept settings data upon submission of settings form and edit user's db entries as appropriate (2-4 hours)
2. (5 story points) User story 1: As an EasyTrage user I want the trade suggestions to take withdrawal fees ~~and my balances~~ into account
 - a. ~~Get user's balances from exchanges every n seconds (2-4 hours)~~
 - b. Use algebra to apply appropriate withdrawal fees to the resulting amount of secondary currency after the first trade, then calculate the resulting primary currency amount after the second trade following the first calculation. Return this as part of the suggestions route and display it on the front-end (2-4 hours)
 - c. ~~Maybe return specific trades per user from the /suggestedTrades route by having the user's JSON web token passed in as an auth header for the request (2-4 hours, maybe unnecessary, can probably just use specific trades per user for the bot only)~~
3. (8 story points) User Story 2: As a user, I want to have the app remember my last deposit address for each exchange and allow me to input the initial amount of the base currency I want to trade for a given suggestion
 - a. Make front-end modal that visualizes trade and displays information from card neatly (4-6 hours)
 - b. Make form for trade and send request to backend to execute trade with amount and deposit address entered. Make request to backend on modal open to populate deposit field with last used deposit address. (2-4 hours)
 - c. Update user's latest deposit address for each exchange on the backend (1 hour)
4. (13 story points) User story 3: As a user, I want to have a bot server-side trade on my behalf
 - a. Make back-end route to take amount user wants to trade and appropriate exchange data from front-end and request a trade (4-8 hours)

- b. Make route to take user secondary currency deposit address on the second exchange of an arbitrage process and transfer the appropriate amount from the resulting value of the first trade on the first exchange (2-4 hours)
- c. ~~Use withdrawal fees to see if moving the USDT/ETH the XRP was sold for to the first exchange again the redo the trade will kill the profit on redoing the trade, if not, loop the trade till it isn't profitable (4-6 hours)~~
- d. ~~Use withdrawal fees to see if user's coins can be moved to another exchange to initiate a good suggested trade without losing the profit from that trade (2-4 hours)~~
- 5. ~~(5 story points) User story 3: As a user, I want to see all my past trades and profits/losses~~
 - a. ~~Whenever the bot trades, append the trade into the user's DB entry with time, prices, etc (1-2 hours)~~
 - b. ~~Make dashboard component to pull and display n recent trades by the bot (or maybe just all) on the user's behalf (2-4 hours)~~

Roles:

- Mike Hamilton: Developer
- Christopher Kong: Developer
- Kevin Lee: Developer
- Vikram Melkote: Product Owner, Developer
- Abdul Mohamed: Scrum Master, Developer

Task Assignment:

- Mike Hamilton: User Story 0: a, b, User Story 1: a, b, User Story 2: a
- Christopher Kong: User Story 0: a, User Story 1: a, b
- Kevin Lee: User Story 2: a, b, c, User Story 3: a, b
- Vikram Melkote: User Story 1: a, b, User Story 2: a, b, c, User Story 3: a, b
- Abdul Mohamed: User Story 1: a, b, User Story 2: a, b, c

Burnup Chart:

